



\*\*FILE\*\* ID\*\*EODEFM

D 11

EEEEEEEEEE 000000 DDDDDDDD EEEEEEEEEE FFFFFFFF MM MM  
EEEEEEEEEE 000000 DDDDDDDD EEEEEEEEEE FFFFFFFF MM MM  
EE 00 00 DD DD EE FF MMMM MMMM  
EE 00 00 DD DD EE FF MMMM MMMM  
EE 00 00 DD DD EE FF MM MM MM MM  
EE 00 00 DD DD EE FF MM MM MM MM  
EE 00 00 DD DD EEEEEEEE FFFFFFFF MM MM  
EE 00 00 DD DD EEEEEEEE FFFFFFFF MM MM  
EE 00 00 DD DD EE FF MM MM MM MM  
EE 00 00 DD DD EE FF MM MM MM MM  
EE 00 00 DD DD EE FF MM MM MM MM  
EE 00 00 DD DD EE FF MM MM MM MM  
EE 000000 DDDDDDDD EEEEEEEE FF MM MM  
EE 000000 DDDDDDDD EEEEEEEE FF MM MM

MM MM AAAAAA RRRRRRRR  
MM MM AAAAAA RRRRRRRR  
MMMM MM AA AA RR RR  
MMMM MM AA AA RR RR  
MM MM AA AA RR RR  
MM MM AA AA RR RR  
MM MM AA AA RRRRRRRR  
MM MM AA AA RRRRRRRR  
MM MM AAAAAA RRR RR  
MM MM AAAAAA R R RR  
MM MM AA AA RR RR

.TITLE EODEF - EDITPC Pattern Operator Macros  
.IDENT V04-000  
.NLIST

\*\*\*\*\*  
COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

\*\*\*\*\*  
FACILITY :  
EDITPC Pattern Operator Encodings

ABSTRACT :  
The EDITPC instruction, Edit Packed to Character String, performs the  
editing according to the pattern string which consists of one byte  
pattern operators. Some pattern operators take no operands. Some  
take a repeat count which is contained in the rightmost nibble of the  
pattern operator itself. The rest take a one byte operand which  
follows the pattern operator immediately. This operand is either an  
unsigned integer length or byte character. Edit patterns must end  
with the EO\$END pattern operator.

The EODEF macros permit easy construction of the edit pattern.

ENVIRONMENT:

AUTHOR: R. P. Grosso, Creation Date 9-Oct-1980

MODIFIED BY:

```
: EOSEND EDIT PATTERN OPERATOR ENCODING
CALL EOSEND
:
.MACRO EOSEND
.BYTE ^X00
.ENDM EOSEND
; EOSEND encoding

: EOSEND_FLOAT EDIT PATTERN OPERATOR ENCODING
CALL EOSEND_FLOAT
:
.MACRO EOSEND_FLOAT
.BYTE ^X01
.ENDM EOSEND_FLOAT
; EOSEND_FLOAT encoding

: EOSCLEAR SIGNIF EDIT PATTERN OPERATOR ENCODING
CALL EOSCLEAR_SIGNIF
:
.MACRO EOSCLEAR_SIGNIF
.BYTE ^X02
.ENDM EOSCLEAR_SIGNIF
; EOSCLEAR_SIGNIF encoding

: EOSSET SIGNIF EDIT PATTERN OPERATOR ENCODING
CALL EOSSET_SIGNIF
:
.MACRO EOSSET_SIGNIF
.BYTE ^X03
.ENDM EOSSET_SIGNIF
; EOSSET_SIGNIF encoding

: EOSSTORE SIGN EDIT PATTERN OPERATOR ENCODING
CALL EOSSTORE_SIGN
:
.MACRO EOSSTORE_SIGN
.BYTE ^X04
.ENDM EOSSTORE_SIGN
; EOSSTORE_SIGN encoding

: EOSLOAD FILL EDIT PATTERN OPERATOR ENCODING
CALL EOSLOAD_FILL <CH>
WHERE CH IS THE FILL CHARACTER
:
.MACRO EOSLOAD_FILL CH
.BYTE ^X40
; EOSLOAD_FILL encoding
.IF BLANK <CH>
; check to be sure CH isn't blank
.WARN :EOSLOAD_FILL - CHARACTER WAS BLANK OR NOT DELIMITED
.ENDC
.IF IDENTICAL CH,/
; avoid .ASCII /// ... if CH is /
.BYTE ^X2F
.IF FALSE
; enter ASCII for "/" ...
; if CH is not "/" then
.ASCII /CH/
.ENDC
.ENDM EOSLOAD_FILL
; fill char placed in fill register
```

## : EOSLOAD SIGN EDIT PATTERN OPERATOR ENCODING

CALL EOSLOAD SIGN &lt;CH&gt;

WHERE CH IS THE SIGN CHARACTER

.MACRO EOSLOAD\_SIGN CH

.BYTE ^X41

: EOSLOAD SIGN encoding

.IF BLANK &lt;CH&gt;

: check to be sure CH isn't blank

.WARN

:EOSLOAD\_SIGN - CHARACTER WAS BLANK OR NOT DELIMITED

.ENDC

.IF IDENTICAL CH./

: avoid .ASCII /// ... if CH is /

.BYTE ^X2F

: enter ASCII for "/"

.IF FALSE

: if CH is not "/" then

.ASCII /CH/

: sign char placed in sign register

.ENDC

.ENDM EOSLOAD\_SIGN

## : EOSLOAD PLUS EDIT PATTERN OPERATOR ENCODING

CALL EOSLOAD PLUS &lt;CH&gt;

WHERE CH IS THE SIGN CHARACTER WHEN RESULT IS POSITIVE

.MACRO EOSLOAD\_PLUS CH

.BYTE ^X42

: EOSLOAD\_PLUS encoding

.IF BLANK &lt;CH&gt;

: check to be sure CH isn't blank

.WARN

:EOSLOAD\_PLUS - CHARACTER WAS BLANK OR NOT DELIMITED

.ENDC

.IF IDENTICAL CH./

: avoid .ASCII /// ... if CH is /

.BYTE ^X2F

: enter ASCII for "/"

.IF FALSE

: if CH is not "/" then

.ASCII /CH/

: char to be placed in sign register

.ENDC

.ENDM EOSLOAD\_PLUS

## : EOSLOAD\_MINUS EDIT PATTERN OPERATOR ENCODING

CALL EOSLOAD\_MINUS &lt;CH&gt;

WHERE CH IS THE SIGN CHARACTER WHEN RESULT IS NEGATIVE

.MACRO EOSLOAD\_MINUS CH

.BYTE ^X43

: EOSLOAD\_MINUS encoding

.IF BLANK &lt;CH&gt;

: check to be sure CH isn't blank

.WARN

:EOSLOAD\_MINUS - CHARACTER WAS BLANK OR NOT DELIMITED

.ENDC

.IF IDENTICAL CH./

: avoid .ASCII /// ... if CH is /

.BYTE ^X2F

: enter ASCII for "/"

.IF FALSE

: if CH is not "/" then

.ASCII /CH/

: char to be placed in sign register

.ENDC

.ENDM EOSLOAD\_MINUS

## : EOSINSERT EDIT PATTERN OPERATOR ENCODING

CALL EOSINSERT &lt;CH&gt;

WHERE CH IS INSERTED

JP

JP

JP

PHI

```
.MACRO EOSINSERT CH
.BYTE ^X44          ; EOSINSERT encoding
.IF BLANK <CH>      ; check to be sure CH isn't blank
.WARN ;EOSINSERT - CHARACTER WAS BLANK OR NOT DELIMITED
.ENC
.IF IDENTICAL CH,/
.BYTE ^X2F          ; avoid .ASCII /// if CH is /
.IF FALSE           ; enter ASCII for "/" if CH is not "/" then
.ASCII /CH/
.ENC
.ENDM EOSINSERT
```

```
: EOSBLANK ZERO EDIT PATTERN OPERATOR ENCODING
CALL EOSBLANK ZERO LEN
WHERE LEN IS A POSITIVE INTEGER
```

```
.MACRO EOSBLANK_ZERO LEN
.BYTE ^X45          ; EOSBLANK_ZERO encoding
.IF EQUAL LEN
.WARN ;EOSBLANK_ZERO - LENGTH SHOULD NOT EQUAL ZERO
.ENC
.BYTE LEN           ; length to fill with contents of
                     ; fill register if value of source
                     ; string is zero.
.ENC EOSBLANK_ZERO
```

```
: EOSREPLACE_SIGN EDIT PATTERN OPERATOR ENCODING
CALL EOSREPLACE_SIGN LEN
WHERE LEN IS A POSITIVE INTEGER
```

```
.MACRO EOSREPLACE_SIGN LEN
.BYTE ^X46          ; EOSREPLACE_SIGN encoding
.IF EQUAL LEN
.WARN ;EOSREPLACE_SIGN - LENGTH SHOULD NOT EQUAL ZERO
.ENC
.BYTE LEN
.ENC EOSREPLACE_SIGN
```

```
: EOSADJUST_INPUT EDIT PATTERN OPERATOR ENCODING
CALL EOSADJUST_INPUT LEN
WHERE LEN IS A POSITIVE INTEGER
```

```
.MACRO EOSADJUST_INPUT LEN
.BYTE ^X47          ; EOSADJUST_INPUT encoding
.IF LESS_EQUAL LEN
.WARN ;EOSADJUST_INPUT - LENGTH MUST BE GREATER THAN ZERO
.ENC
.IF GREATER_EQUAL LEN - 32
.WARN ;EOSADJUST_INPUT - LENGTH MUST BE LESS THAN 32
.ENC
```

.BYTE LEN : length to fill with contents of  
: fill register if value of source  
: string is zero.

.ENDM EOSADJUST\_INPUT

: EOSFILL EDIT PATTERN OPERATOR ENCODING  
CALL EOSFILL R  
WHERE R, THE REPEAT COUNT IS BETWEEN 1 AND 15 INCLUSIVE

.MACRO EOSFILL R  
.IF LESS\_EQUAL R : repeat cannot be zero  
.WARN ;EOSFILL - REPEAT CANNOT BE ZERO  
.ENDC  
.IF GREATER\_EQUAL R - 16 : repeat count must be contained in  
: a nibble  
.WARN ;EOSFILL - REPEAT IS GREATER THAN 15  
.ENDC  
.BYTE <^X80 + R> : EOSFILL operator encoding  
: plus repeat count

.ENDM EOSFILL

: EOSMOVE EDIT PATTERN OPERATOR ENCODING  
CALL EOSMOVE R  
WHERE R, THE REPEAT COUNT IS BETWEEN 1 AND 15 INCLUSIVE

.MACRO EOSMOVE R  
.IF LESS\_EQUAL R : repeat cannot be zero  
.WARN ;EOSMOVE - REPEAT CANNOT BE ZERO  
.ENDC  
.IF GREATER\_EQUAL R - 16 : repeat count must be contained in  
: a nibble  
.WARN ;EOSMOVE - REPEAT IS GREATER THAN 15  
.ENDC  
.BYTE <^X90 + R> : EOSMOVE operator encoding  
: plus repeat count

.ENDM EOSMOVE

: EOSFLOAT EDIT PATTERN OPERATOR ENCODING  
CALL EOSFLOAT R  
WHERE R, THE REPEAT COUNT IS BETWEEN 1 AND 15 INCLUSIVE

.MACRO EOSFLOAT R  
.IF LESS\_EQUAL R : repeat cannot be zero  
.WARN ;EOSFLOAT - REPEAT CANNOT BE ZERO  
.ENDC  
.IF GREATER\_EQUAL R - 16 : repeat count must be contained in  
: a nibble  
.WARN ;EOSFLOAT - REPEAT IS GREATER THAN 15  
.ENDC  
.BYTE <^XA0 + R> : EOSFLOAT operator encoding  
: plus repeat count

.ENDM EOSFLOAT

EODEFM.MAR;1

16-SEP-1984 17:07:43.38 Page 6

:+  
:- END OF EDIT PATTERN OPERATOR ENCODINGS

0434 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY